IN THE CLAIMS

1. (Currently amended) A data processing system for executing a program of virtual machine instructions with a processor core that is arranged to execute native instructions comprising

the processor core;

a memory;

a virtual machine interpreter, comprising a hardware pre-processor including a program counter, separate from the processor core, for receiving virtual machine instructions selected dependent on program flow during execution of the program, the virtual machine interpreter being coupled to the processor core to generate native machine instructions that implement the virtual machine instructions for execution by the processor core, the virtual machine interpreter being arranged

to identify an initial virtual machine instruction from a body of successive ones of the selected virtual machine instructions, where the body is expected to be executed repeatedly; to record a correspondence between the initial virtual machine instruction in the body and a memory location in the memory;

to write native instructions for the body into the memory from said memory location, the native instructions for the body being generated for virtual machine instructions starting from the initial virtual machine instruction;

to cause the processor core to execute the native instructions for the body and

to repeat execution of the native instructions for the body by executing the written native machine instructions for the body from memory starting from said memory location.

2. (Previously presented) A data processing system according to claim 1, the virtual machine interpreter being arranged to generate a native branch back instruction to the a start of the body and placing the native branch back instruction at the end of the body in the memory.

3. (Previously presented) A data processing system according to claim 2, the

virtual machine interpreter being arranged to place an unconditional further native branch instruction behind the native branch instruction, the unconditional further native branch instruction having a target address in a range of addresses that does not overlap a further range of addresses in which the body is stored, the virtual machine interpreter being arranged to monitor a program counter address of the processor core and to resume selection of the virtual machine instructions and generation of native machine instructions from the selected virtual machine instructions when the program counter address enters said range of addresses after execution of the loop body.

4. (Previously presented) A data processing system according to claim 1, the virtual machine interpreter being arranged to receive hint information, which does not affect program flow, the hint information indicating at least said initial virtual machine instruction, the virtual machine interpreter recording said correspondence and writing the native instructions for the body when program flow reaches the initial virtual machine instruction, conditional upon receiving said hint information.

5. (Currently amended) A method of executing a program of virtual machine instructions with a processor core that is arranged to execute native instructions, the method comprising

selecting, under control of program flow, virtual machine instructions to be executed;

using a hardware pre-processor, determining native instructions from the selected virtual machine instructions, to implement the selected virtual machine instructions;

for a body of successive ones of the selected virtual machine instructions that is expected to be executed repeatedly,

identifying an initial virtual machine instruction of the body of successive ones of the selected virtual machine instructions;

recording a correspondence between the initial virtual machine instruction and a memory location;

writing native instructions for the body into a memory from said memory location, the native instructions for the body being determined from virtual machine instructions starting from the initial virtual machine instruction;

causing the processor core to execute the native instructions for the body and to repeat execution of the native instructions for the body by executing the written native machine instructions for the body from memory starting from said memory location; and

for selected virtual machine instructions that are not expected to be executed repeatedly, providing directly to the processor core instructions to implement the selected virtual machine instructions, without storing the instructions to implement the selected virtual machine instructions in memory.

6. (Previously presented) A method according to claim 5, comprising generating a native branch back instruction to a start of the body and placing the native branch back instruction in the memory at the end of the body of successive ones of the selected virtual machine instructions that is expected to be executed repeatedly.

7. (Previously presented) A method according to claim 6, comprising placing an unconditional further native branch instruction behind the native branch instruction, the unconditional further native branch instruction having a target address in a first range of addresses that does not overlap a further range of addresses in which the body is stored, the method comprising the step of monitoring a program counter address of the processor core and to resume said selecting and determining when the program counter address enters said first range of addresses after execution of the body of successive ones of the selected virtual machine instructions that is expected to be executed repeatedly.

8. (Previously presented) A method according to claim 5, said identifying comprising preprocessing the program to detect loop terminating with a virtual machine branch

back instruction and adding a hint to the program which identifies a target address of the virtual machine branch back instruction as the initial virtual machine instruction.